# Data Structures Study Guide

**Introduction to Data Structures**
Data structures are specialized formats for organizing, processing, storing, and retrieving data. Choosing the right data structure is crucial for writing efficient programs.

**1. Arrays**
- Fixed-size sequential collection of elements
- Time Complexity: Access O(1), Search O(n), Insert/Delete O(n)
- Use cases: When size is known, random access needed

**2. Linked Lists**
- Linear collection of nodes containing data and reference to next node
- Types: Singly, Doubly, Circular
- Time Complexity: Access O(n), Insert/Delete O(1) at known position
- Use cases: Dynamic size, frequent insertions/deletions

**3. Stacks**
- LIFO (Last In First Out) structure
- Operations: Push, Pop, Peek
- Time Complexity: O(1) for all operations
- Use cases: Function calls, undo operations, expression evaluation

**4. Queues**
- FIFO (First In First Out) structure
- Operations: Enqueue, Dequeue, Front
- Time Complexity: O(1) for all operations
- Use cases: Task scheduling, breadth-first search

**5. Trees**
- Hierarchical structure with root and children
- Binary Tree: Each node has at most two children
- Binary Search Tree: Left child < parent < right child
- Use cases: File systems, databases, decision making

**6. Hash Tables**
- Key-value pairs with hash function for fast lookup
- Time Complexity: Average O(1) for insert, delete, search
- Use cases: Caching, databases, counting frequencies

**7. Graphs**
- Collection of nodes (vertices) and edges
- Types: Directed, Undirected, Weighted
- Use cases: Social networks, maps, recommendation systems

**Study Tips:**
- Understand time and space complexity
- Practice implementing each structure from scratch
- Know when to use which data structure
- Practice common interview problems for each structure